KEKER & VAN NEST LLP
ROBERT A. VAN NEST - #84065
rvannest@kvn.com
CHRISTA M. ANDERSON - #184325
canderson@kvn.com
633 Battery Street
San Francisco, CA 94111-1809
Telephone: 415.391.5400
Facsimile: 415.397.7188

KING & SPALDING LLP
DONALD F. ZIMMER, JR. - #112279
fzimmer@kslaw.com
CHERYL A. SABNIS - #224323
csabnis@kslaw.com
101 Second St., Suite 2300
San Francisco, CA 94105
Telephone: 415.318.1200
Facsimile: 415.318.1300

KING & SPALDING LLP
SCOTT T. WEINGAERTNER (*Pro Hac Vice)*
sweingaertner@kslaw.com
ROBERT F. PERRY
rperry@kslaw.com
BRUCE W. BABER (*Pro Hac Vice)*
1185 Avenue of the Americas
New York, NY 10036
Telephone: 212.556.2100
Facsimile: 212.556.2222

GREENBERG TRAURIG, LLP
IAN C. BALLON - #141819
ballon@gtlaw.com
HEATHER MEEKER - #172148
meekerh@gtlaw.com
1900 University Avenue
East Palo Alto, CA 94303
Telephone: 650.328.8500
Facsimile: 650.328.8508

Attorneys for Defendant
GOOGLE INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Case No. 3:10-cv-03561-WHA

DEFENDANT GOOGLE INC.'S NOTICE OF MOTION AND MOTION FOR SUMMARY JUDGMENT ON COUNT VIII OF PLAINTIFF ORACLE AMERICA'S AMENDED COMPLAINT

Judge: Hon. William Alsup

Hearing: 2:00 p.m., September 15, 2011

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

II. FACTUAL BACKGROUND 2

A. A brief history of Java and Android. 2

1. The Java language and the Java platform. 2

2. The Android platform. 4

B. APIs define how pieces of software can interact with each other. 6

C. An example: How to calculate an absolute value in Java. 7

D. Oracle's copyright infringement claim. 9

III. ARGUMENT 11

A. Because Oracle cannot establish that Google copied *protected* elements of the Java language API specifications, the Android APIs are not substantially similar to Oracle's Java language API specifications 11

1. The only elements common to Oracle's Java language APIs and the Android APIs are unprotectable methods of operation. 12

2. The API declarations are unprotectable *scenes a faire* or unprotectable under the merger doctrine. 14

3. The Java language API package and method names are unprotectable as a matter of law. 17

4. Alternatively, any similarity between the works is a fair use. 19

B. The alleged similarities in the remaining 12 files are *de minimis* in the context of the over *9,500* files in the Asserted Works, and the over 50 *thousand* files in Android. 22

C. The documentation for the Android APIs is not substantially similar or virtually identical to the documentation for the Java language APIs. 24

D. Any claim based on Oracle's selection and arrangement of allegedly copied elements must be evaluated under the "virtual identity" standard, and Oracle cannot establish infringement under that standard 24

E. Oracle's secondary infringement claims fail for lack of proof of direct infringement. 25

IV. CONCLUSION 25

TABLE OF AUTHORITIES

Page(s)

Federal Cases

Apple Computer, Inc. v. Microsoft Corp.
35 F.3d 1435 (9th Cir. 1994) 11, 12, 24, 25

Baker v. Selden
101 U.S. 99 (1879).....12

Bateman v. Mnemonics, Inc.
79 F.3d 1532 (11th Cir. 1996) 15, 19

Baystate Techs. v. Bentley Sys.
946 F. Supp. 1079 (D. Mass. 1996).....15, 16, 18

Brown Bag Software v. Symantec Corp.
960 F.2d 1465 (9th Cir. 1992) 4

CMM Cable Rep, Inc. v. Ocean Coast Props.
97 F.3d 1504 (1st Cir. 1996).....17

Computer Assocs. Int'l, Inc. v. Altai
982 F.2d 693 (2d Cir. 1992) 14, 15, 18, 22

Data East USA, Inc. v. Epyx, Inc.
862 F.2d 204 (9th Cir. 1988) 11

Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.
499 U.S. 340 (1991).....11, 14

Fisher v. Dees
794 F.2d 432 (9th Cir. 1986) 23

Jada Toys, Inc. v. Mattel, Inc.
518 F.3d 628 (9th Cir. 2008) 11

Kelly v. Arriba Soft Corp.
336 F.3d 811 (9th Cir. 2003) 19, 21, 22

Lexmark Int'l., Inc. v. Static Control Components, Inc.
387 F.3d 522 (6th Cir. 2004) 15

Lotus Dev. Corp. v. Borland Int'l, Inc.
49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided court,*
516 U.S. 233 (1996).....2, 4, 12, 13, 14

Matthew Bender & Co. v. West Publ'g Co.
158 F.3d 693 (2d Cir. 1998) 25

Merchant Transaction Sys., Inc. v. Nelcela, Inc.
No. CV 02-1954-PHX-MHM, 2009 WL 723001
(D. Ariz. March 18, 2009) 17

MiTek Holdings, Inc. v. Arce Eng'g Co., Inc.
864 F. Supp. 1568 (S.D. Fla. 1994), *aff'd*, 89 F.3d 1548
(11th Cir. 1996).....24

Mitel, Inc. v. Iqtel, Inc.
896 F. Supp. 1050 (D. Colo. 1995), *aff'd*, 124 F.3d 1366
(10th Cir. 1997).....13, 14, 15, 16, 19

TABLE OF AUTHORITIES
(cont'd)

Page(s)

Newton v. Diamond
388 F.3d 1189 (9th Cir. 2004) 22, 23

Perfect 10, Inc. v. Amazon.com, Inc.
508 F.3d 1146 (9th Cir. 2007) 19, 22

Sega Enters. Ltd. v. Accolade, Inc.
977 F.2d 1510 (9th Cir. 1992) ..... *passim*

Sony Computer Entm't, Inc. v. Connectix Corp.
203 F.3d 596 (9th Cir. 2000) 20, 21, 22

Subafilms, Ltd. v. MGM-Pathe Communications Co.
24 F.3d 1088 (9th Cir. 1994) 25

Federal Statutes

17 U.S.C. § 102(b) 11, 12, 13

Federal Regulations

37 C.F.R. § 202.1(a) 17

Constitutional Provisions

U.S. CONST. art. I, § 8, cl. 8 14

Other Authorities

Andrew Sinclair, *License Profile: Apache License, Version 2.0* 4

Paul GOLDSTEIN, GOLDSTEIN ON COPYRIGHT § 2.15.3, at 2:208
(3d ed. 2011) 16, 17

NOTICE OF MOTION AND STATEMENT OF RELIEF SOUGHT

PLEASE TAKE NOTICE, that on September 15, 2011, at 2:00 p.m., or at such other time as the Court may direct, before the Honorable William Alsup, United States District Court, 450 Golden Gate Avenue, San Francisco, California 94102, Defendant Google Inc. ("Google") will, and hereby does, move the Court for entry of summary judgment on Count VIII of the Amended Complaint filed by Plaintiff Oracle America, Inc. ("Oracle").

This Motion is based on this Notice of Motion and Motion, the Memorandum of Points and Authorities below, the Declarations of Daniel Bornstein, Owen Astrachan and Michael S. Kwun that are being filed herewith, and such other and further papers, evidence and argument as may be submitted to the Court in connection with the hearing on this motion.

MEMORANDUM OF POINTS AND AUTHORITIES

I. INTRODUCTION

In 1994, the Chief Technology Officer ("CTO") of Sun Microsystems ("Sun")[1] offered the following testimony to Congress: "With respect to intellectual property rights, Sun strongly believes in—and will defend—the rights of intellectual property owners to maximize their returns on product implementations. At the same time, we believe that *interface specifications are not protectable under copyright.*"[2]

Yet Oracle's copyright claim rests almost entirely on the use of the interface specifications (specifications for application programming interfaces, or APIs) in thirty-seven[3] Java language API packages in Android. Those specifications are "functional requirements for compatibility . . . that are not protected by copyright." *See Sega Enters. Ltd. v. Accolade, Inc.*,

---

[1] Sun is Oracle's predecessor in interest.

[2] Declaration of Michael S. Kwun, Ex. G at 2 (emphasis added). The cited exhibit includes the entirety of the 1994 prepared testimony of Dr. Eric Schmidt, testifying in his capacity as Sun's CTO.

[3] In February, arguing that a motion for summary judgment would be premature, Oracle said it expected discovery would reveal further instances of alleged copying other than those of which it was then aware. *See* Kwun Decl, Ex. N at 1. Today, more than six months later, the only thing that has changed is that the number of API packages on which Oracle bases its infringement claim has *declined* from fifty-one, *see id.*, Ex. O at 7:4-9:1, to thirty-seven, *see id.*, Ex. C at 10:1-11:9.

977 F.2d 1510, 1522 (9th Cir. 1992) (citing 17 U.S.C. § 102(b)). They define how programmers interact with the Android platform, much as the menus in a computer program define how users interact with the program. And just as menus in a computer program are not protected by copyright, *see Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S. 233 (1996), API specifications are not copyrightable. What the specifications *do not* do is define how the Android platform actually does the things it does, just as the "print" menu option in a computer program is different from the code that actually causes the program to print a document.

That the Java language API *specifications* are not themselves copyrightable presents a pure question of law. As Sun's CTO explained in 1994, "The interface, as an element necessary for interoperability, falls into the category of ideas which the copyright law seeks to disseminate to promote the public good. This in no way curtails the protectability of the code itself."[4]

Aside from the Java language API specifications, Oracle's copyright claim is based on minimal alleged copying, concerning only 12 files out of over 50 *thousand* in Android, and material of no qualitative significance. A reasonable jury could only conclude that any such similarities are *de minimis* and thus not actionable. Google is therefore entitled to summary judgment on the entirety of Oracle's copyright claim.

II. FACTUAL BACKGROUND

A. A brief history of Java and Android.

1. The Java language and the Java platform.

Oracle uses "Java platform" to mean a variety of interchangeable and overlapping elements. These elements purportedly include the Java programming language itself, an "object-oriented" programming language that uses syntax heavily based on prior languages such as the "C" and "C++" programming languages. Declaration of Owen Astrachan, Ex. 1 ¶ 18. However, "it is undisputed that the Java programming language is in the public domain and anyone was free to use it without charge, as Android does." Kwun Decl., Ex. F (July 22, 2011 Order) at 2;

[4] Kwun Decl., Ex. G at 2.

see also id., Ex. E (Feb. 9, 2011 Hearing Tr.) at 8:17-18 ("[W]e're not asserting that we own [the Java] programming language for purposes of this case."). Also included in the elements of the Java platform are a program known as a "compiler" that creates the "bytecode" in which Java programs are executed; a virtual machine that executes the bytecode; and a set of core libraries that facilitates the development of applications for the Java platform by providing basic system or language functionalities. Astrachan Decl., Ex. 1 ¶ 18.

Today, Java is a popular programming language, and many Internet and mobile applications are written in the Java language. *Id.* Indeed, Oracle claims that Java has attracted 6.5 million software developers, who must be familiar with, and accustomed to, the naming conventions and functionalities of the basic Java language to develop their programs. *See* Kwun Decl., Ex. A (Am. Compl.) ¶ 9.

The Java programming language was designed to use a grammar and syntax that was similar to those of other well-known, existing languages, such as C and C++. Astrachan Decl., Ex. 1 ¶ 124. Reusing C and/or C++ grammar and syntax allowed Sun to leverage the existing knowledge of programmers at the time the Java programming language was developed. *Id.* As the primary creator of the Java language has acknowledged, "Java feels very familiar to many different programmers because Sun had a very strong tendency to prefer things that had been used a lot over things that just sounded like a good idea." *Id.*, Ex. 1 ¶ 124 & Ex. 2 at 53.

Like any high-level programming language, the Java programming language contains many rules of grammar and syntax that generally cannot be varied. *Id.*, Ex. 1 ¶ 19. For example, a statement adding two numbers can only be written in certain ways, and the language requires specific and precise keywords to express such things as variable types (integers, strings, or booleans) and more complex object types such as dates or database queries. *Id.* Even the capitalization of identifiers in the Java programming language is subject to language requirements and best practices. *Id.* ¶ 113. In addition, the Java language, like many programming languages, employs keywords and operators (such as plus and minus symbols) that can only be used for specific purposes and in specific ways; using them for other purposes will cause a program to fail to function correctly. *Id.* ¶ 19. Much of the structure and appearance of

code written in the Java programming language is therefore *dictated by functional considerations*. *Id.*

2. The Android platform.

The Android platform is a freely-distributed, open-source software stack for mobile devices that includes an operating system, middleware and key mobile applications. Declaration of Daniel Bornstein ¶ 2. In the early days of its development, Android, Inc. and then Google (which purchased Android, Inc. in 2005) pursued numerous possibilities for developing the platform, including a technology partnership with Sun (incorporating Sun's implementation of its virtual machine and Java language libraries into the platform), as well as developing Android from the ground up. *Id.* When the partnership negotiations failed, Google decided to concentrate fully on the ground up solution, using only new materials and publicly-available open-source materials. *Id.*

Android was built to be a truly "open" platform. Android is made available under permissive open source license terms, primarily the Apache License, Version 2.0. *Id.* ¶ 3.[5] The information and source code for the Android platform are freely available for developers, manufacturers, or any member of the general public to download at http://source.android.com and http://developer.android.com. *Id.* The software development kit ("SDK") for the Android platform was first publicly released by Google in 2007. *Id.*

Android provides access to a wide range of useful libraries and tools that can be used to build rich mobile applications by interfacing with hardware that is typically not available on desktop platforms. *Id.* ¶ 4. For example, Android enables developers to obtain the location of the mobile device using the phone's built-in GPS technology, or to create photography applications using the built-in camera. *Id.* Developers for Android can create software applications for Android-based mobile devices using various programming languages, including the Java programming language. *Id.* Other languages supported by Android include C, C++, and JavaScript, among others. *Id.*

[5] *See generally* Andrew Sinclair, *License Profile: Apache License, Version 2.0*, 2 INT'L FREE & OPEN SOURCE SOFTWARE L. REV. 107 (2010) (discussing Apache License, Version 2.0).

Android contains over 50 thousand files and over 11 million lines of code, and is built on an underlying Linux kernel. *Id.* ¶ 5 The Linux kernel is an open source operating system kernel that communicates directly with the hardware device and manages core functions and communications (such as those relating to memory management, process management and hardware drivers) between the applications and the device. *Id.* ¶ 5 & Ex. A (diagram illustrating the major components of the Android platform).

The Android platform includes, among other things, the Android SDK and the Dalvik Virtual Machine ("VM"). *Id.* ¶ 6. The Android SDK is a comprehensive set of development tools provided to software developers who wish to create applications for Android devices. *Id.* The Dalvik VM is a new intermediate software layer, created by Google, that allows programs, including those written in the Java programming language, to run on the Android platform. *Id.* The Dalvik VM and Android platform accomplish this function in part by using a collection of 168 libraries (also referred to as "packages") referred to as the "Android Core Libraries." *Id.*

Some of the Android Core Libraries incorporate and were in part derived from Java language API libraries from Apache Harmony ("Harmony" or the "Harmony Project"). *Id.* ¶ 7. The Harmony Project, founded in May 2005, is an open source project that includes libraries written to be interoperable with the Java language APIs included in the Java 2 Standard Edition platform. *See* Kwun Decl., Exs. H at 2 & I at 1. Harmony, which predates Android, was developed by the non-profit Apache Software Foundation. The Harmony Project took steps from its creation to ensure that Sun's (now Oracle's) intellectual property was respected, calling the issue "very important" in the project's initial announcement and detailing three separate mechanisms by which the project would ensure respect of Sun's rights. *See id.*, Ex. J at 3 (text under the heading "How will you ensure that the intellectual property of Sun and others is respected?"). To the best of Google's knowledge, neither Sun nor Oracle has ever accused the Apache Software Foundation or its libraries of infringing any of Oracle's copyrights.[6]

As part of the early development work on Android, Google retained an independent

[6] Harmony was not the first open source implementation of common Java language APIs. An earlier effort, now called GNU Classpath, was launched in 1998. *See id.*, Ex. K at 2.

contractor to develop certain Android Core Libraries. Bornstein Decl. ¶ 8. Google informed the contractor that it was interested in interoperability with version 1.5 (now known as version 5.0) of the Java 2 Standard Edition platform. *Id.* In other words, Google—like Apache—was interested in maximizing compatibility with the Java programming language. The contractor was retained and instructed to write new code, reuse and improve original code created by Google, or import and improve code from permissively licensed (or equivalent) open source projects, such as the Harmony Project—but not from any proprietary materials, including Sun materials. *Id.*

B. APIs define how pieces of software can interact with each other.

An API allows one piece of software to call on another piece of software to perform a function; in other words, it is a method or means through which two computer programs or portions of programs are able to interact and communicate with each other. Astrachan Decl., Ex. 1 ¶ 24; *see also id.* ¶¶ 26-35. An API is typically defined in a written document known as a "specification," which describes the API and its function and explains how the API is to be implemented in code. *Id.* ¶ 55. The specification gives the name of the API, the required inputs for the API, the expected results of invoking the API, and the requirements that the code implementing the API (including any associated files or libraries needed for the API to execute properly) must meet in order to carry out the purpose of the API correctly. *Id.* The specifications for the thirty-seven Java language API packages at issue were published or made available in various forms, including in books and on the Java website, starting in 1996. *Id.* ¶ 20.

Much like many prior programming languages, Java language APIs also typically include subparts or files known as "classes" and, within classes, "methods." *Id.* ¶¶ 36-38, 48. Each of the Java language API packages at issue is a grouping of classes, which themselves are groupings of methods. *Id.* ¶ 48. The source code for the APIs for the Java programming language typically includes text known as "comments"; comments do not affect the compiled code but, instead, may document what the code does. *See id.* ¶¶ 58, 171. For example, the comments may describe to the programmer the function and rules for using the API. *Id.* ¶ 58. For Java, the reference materials used by programmers (typically referred to as the API's

"documentation") is machine-generated automatically from these comments. *See id.*

C. An example: How to calculate an absolute value in Java.

One of the API packages that Oracle accuses Google of infringing is the "java.lang" package. Kwun Decl. Ex. C at 10:4. According to Oracle's documentation, this package "[p]rovides classes that are fundamental to the design of the Java programming language." Astrachan Decl., Ex. 1 ¶ 39. One of those "fundamental" classes is the "Math" class, which "contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions." *Id.* One such method defined in the Math class is the "abs" method, which is used to calculate the absolute value for a given number. *Id.*

The "integer" version of the abs method[7] has the "declaration"[8] "public static int abs (int i)." *Id.* ¶ 57. The declaration identifies the method and uses the name of the method set forth in the specification. *Id.* ¶¶ 40-46. The "int i" in the parentheses means that the abs method accepts one "argument" as an input, and the input is an integer ("int" is short for "integer"). *Id.* ¶ 47. The "static int" before the "abs" in the declaration means that this version of the abs method returns an integer. *Id.* A programmer who wants to calculate the absolute value of, for example, -25 would thus include the text "abs(-25)" in his or her program. *Id.* ¶ 37.

The Android code for the "abs" method discussed above is in the Android file Math.java. *Id.* ¶ 57 n.2. The portion of this file relevant to the abs method is:

```
package java.lang;
  . . .
    /**
     * Returns the absolute value of the argument.
     * <p>
     * If the argument is {@code Integer.MIN_VALUE}, {@code
Integer.MIN_VALUE}
     * is returned.
     *
     * @param i
     *            the value whose absolute value has to be computed.
     * @return the argument if it is positive, otherwise the negation
```

[7] *See id.* ¶ 36 (explaining what a "method" is). There are three versions of the abs method, which are used for calculating the absolute value of integers, floating point numbers (such as 1.23), and double precision floating point numbers.

[8] *See id.* ¶¶ 40-46 (explaining what a method "declaration" is).

```
     of the
          *          argument.
          */
     public static int abs(int i) {
          return i >= 0 ? i : -i;
     }
```

Id. ¶ 57. The first line, "package java.lang," means that this file is part of the "java.lang" package. *Id.* This is the name of the package of classes in which this file resides. *Id.* The text beside the asterisks in each line are comments embedded in the code (shown in blue) that explain this method to programmers, which (as discussed above) in turn are used to automatically generate the "documentation" for the method. *Id.* ¶ 58. The line "public static int abs(int i)" and the lines beneath it repeat the declaration of the "abs" method (shown in green) and then present the program logic for the method (shown in red). *Id.* ¶ 59.

In this example, the actual *program logic* of the method—the code that actually calculates the absolute value—is the text "return i >= 0 ? i : -i;", which effectively says, "if the number we are given is greater than or equal to 0, return that number, and otherwise return that number but with the opposite sign." *Id.* Because the absolute value operation is relatively simple, very little program logic is needed for this method. *Id.* For more complicated methods, many lines of program logic may be needed. *Id.* Everything *other* than this program logic describes what the method is called, how a programmer can invoke it, what the method does, and what the method will return when it is completed. The program logic for the method, in contrast, is the code that actually tells the computer *how* to perform the method (in this case, how to calculate the absolute value of an integer).

The "abs" method declaration is the same not only in Java and Android, but also in the two major open source Java language projects—Apache Harmony and GNU Classpath. *Id.* ¶ 61. All four use identical method declarations:

Java:	public static int abs(int a)
Harmony:	public static int abs(int i)
GNU Classpath:	public static int abs(int i)
Android:	public static int abs(int i)

Id. In addition, both Apache Harmony and GNU Classpath have implemented the Java language API packages that are the subject of Oracle's infringement claim against Google. *Id.* In each case, the names of the packages and methods are the same. *Id.*

D. Oracle's copyright infringement claim.

In its interrogatory answers, Oracle has set forth the basis for its claim of copyright infringement. *See* Kwun Decl., Ex. C (relevant excerpts from Oracle's Supplemental Responses to Google's First Set of Interrogatories, served on July 29, 2011). Oracle alleges in its Responses that the *specifications* for thirty-seven Android Java language API packages (the "Accused Packages") "are unauthorized derivative work" and are "derived from or substantially similar to" the corresponding Java language API specifications contained in the Oracle's asserted works.[9] *See id.* at 9:15-11:9. Oracle points to no specific code in the API packages that it claims to have been copied from Oracle code; rather, it asserts that Google has infringed by including in Android implementations of APIs that provide compatibility for applications written in the standard Java programming language. *See id.* at 11:10-14. Indeed, Oracle's illustrative exhibits that purportedly show the infringement do not include *any* of the program logic for any of the methods. *See* Kwun Decl., Ex. D (exhibits to Oracle's Responses).

Although Oracle also claims that the Android code that *implements* those specifications is an "unauthorized derivative work," Kwun Decl. Ex. C at 11:16, when it comes to the actual material that allegedly was copied, Oracle again points only to its *specifications*. *Id.* at 12:22-24 ("For example, Google makes and distributes dalvik\vm\native\java_lang_Class.c, which is based on Oracle's java.lang.Class specification").[10]

---

[9] Oracle's Amended Complaint pleads only two copyright registrations. *See* Kwun Decl., Exs. A (Am. Compl.) ¶ 11 & B (Am. Compl., Ex. H). Those registrations correspond to "Java 2 Standard Edition 1.4" and "Java 2 Standard Edition, Version 5.0." These two works are referred to as the "Asserted Works." Oracle also referred to additional copyright registrations in an interrogatory response, *see* Kwun Decl., Ex. C at 9:11-13, but has not sought to further amend its complaint beyond the Asserted Works. That said, nothing in Oracle's interrogatory responses suggests that the analysis of its copyright claim would change based on these additional works.

[10] Essentially the same text, again without references to anything beyond alleged similarities in the *specifications* for the APIs, appears in Oracle's response to Google's interrogatory seeking identification of the specific material that Oracle alleges was copied from the Asserted Works. *See* Kwun Decl., Ex. C at 20:5-22:25.

The only allegations of literal code copying specifically identified by Oracle relate to 12 specific files (the "Accused Files")[11]—out of more than 50 *thousand* files in Android—that contain portions that Oracle alleges are similar to Oracle code. *See id* at 13:26-14:27. This group of files, in the aggregate, comprises far less than one percent of Android, whether analyzed in terms of the number of files or the number of lines of code. Astrachan Decl., Ex. 1 ¶ 150. These twelve files also amount to far less than one percent of the Asserted Works. *Id.*

Eight of the twelve files identified by Oracle,[12] moreover, came from a Google contractor, and are "test files" for the purpose of verifying that certain aspects of the corresponding program file worked correctly. Bornstein Decl. ¶ 8. These files do not have any effect on the code that ships on Android devices, and there is no reason that these files would ever be present on Android devices. *Id.* In fact, after this lawsuit was filed, Google removed these unnecessary files from the Android platform, and has not replaced them with new files. *Id.*

The other four files include minimal text similar to text in Oracle materials. The only allegedly copied text in two of those files[13]—files that Google received from the same independent contractor that provided it with the eight files just discussed, *id.* ¶ 9—is in twenty "comments," i.e. text that documents the code rather than operative code itself. Astrachan Decl., Ex. 1 ¶ 170. Those unnecessary comments have also been removed from the Android files. Bornstein Decl. ¶ 9.[14] The final two files[15] each include only nine (out of several hundred) lines

---

[11] In its response to a Google interrogatory (Interrogatory No. 2), Oracle has specifically identified the twelve files that it contends contain code copied from Oracle code. *See id.* at 13:26-14:27. In response to a further interrogatory (Interrogatory No. 5) seeking the specific copied code, Oracle has steadfastly refused to identify the specific code that it contends was copied, instead only repeating essentially the same general statements contained in its response to the other interrogatory. *See id.* at 20:5-25:10.

[12] In Oracle's Responses, these files: AclEntryImpl.java; AclImpl.java; GroupImpl.java; OwnerImpl.java; PermissionImpl.java; PrincipalImpl.java; PolicyNodeImpl.java; and AclEnumerator.java. *See* Kwun Decl., Ex. C at 14:1-17.

[13] In Oracle's Responses, these files: CodeSourceTest.java; and CollectionCertStoreParametersTest.java. *See* Kwun Decl., Ex. C at 14:22-27.

[14] These files were appear to drive from Apache Harmony files, and the allegedly "copied" comments were apparently added to the Apache Harmony files by Intel in 2006. Astrachan Decl. ¶¶ 174-75. Google received these files from a contractor. Bornstein Decl. ¶ 9.

[15] In Oracle's Responses, these files: TimSort.java; and ComparableTimSort.java. *See* Kwun Decl., Ex. C at 14:18-21.

that appear to be the same as lines in the Oracle materials. Astrachan Decl., Ex. 1 ¶ 152. Those nine lines (which are the same in both of the Android files) implement a mundane utility function. *Id.* ¶¶ 153-56.

III. ARGUMENT

A. Because Oracle cannot establish that Google copied *protected* elements of the Java language API specifications, the Android APIs are not substantially similar to Oracle's Java language API specifications.

To establish copyright infringement, a plaintiff must prove ownership of the copyrighted work and that the defendant copied *protected* elements of the work. *Jada Toys, Inc. v. Mattel, Inc.*, 518 F.3d 628, 636 (9th Cir. 2008). "The mere fact that a work is copyrighted does not mean that every element of the work may be protected." *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 348 (1991). Ordinarily, to prove copying, a plaintiff must show that defendant "had access to the copyrighted work and that the protected portions of the works are substantially similar." *Jada Toys*, 518 F.3d at 636-37.

Copyright protects only the expression of an idea, not the idea itself. 17 U.S.C. § 102(b). Where the similarities between a plaintiff's and defendant's works are limited to "ideas and general concepts," there is no infringement. *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 208 (9th Cir. 1988). Substantial similarity cannot be based on expression where an idea and its expression are inseparable, or where that expression is "as a practical matter, indispensable or at least standard in the treatment of a given [idea]." *Id.* (quotation marks and citations omitted). To avoid an improper finding of infringement based on unprotectable expression, Ninth Circuit law calls for an "analytic dissection" of similarities. *Id.* Where all similarities in expression arise from unprotectable elements, there is no substantial similarity, and thus no infringement. *Id.* Analytic dissection to determine the proper scope of a copyright is a question of law. *See Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 (9th Cir. 1994) ("Using analytic dissection, and, if necessary, expert testimony, *the court* must determine whether any of the allegedly similar features are protected by copyright" (emphasis added)).

As explained below, the only similarities between the Accused Packages and the Java language API packages lie in the specifications for those APIs. But similarities in only

"functional specifications," as opposed to the program logic that performs the steps required by those specifications, are not copyright infringement. *See Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1526 (9th Cir. 1992). Moreover, where, as here, "the range of protectable and unauthorized expression is [at most] narrow, the appropriate standard for illicit copying is virtual identity." *Apple*, 35 F.3d at 1439. Because Oracle cannot show that the Android platform is virtually identical to the Asserted Works (or, indeed, even substantially similar), Google is entitled to summary judgment that the Accused Packages do not infringe the Asserted Works.

1. The only elements common to Oracle's Java language APIs and the Android APIs are unprotectable methods of operation.

As long ago as 1879, the Supreme Court made clear that publication of a book that explains a particular accounting system gives the author no rights under the copyright laws to prevent others from *using* the system, as long as no protectable expression from the book is copied. *Baker v. Selden*, 101 U.S. 99 (1879); *see also* 17 U.S.C. § 102(b) ("In no case does copyright protection . . . extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."). Oracle's Java language books and API specifications give Oracle no greater rights to prevent implementation by others of the APIs using original code.

This principle applies equally in the computer context. In *Lotus Development Corp. v. Borland Int'l, Inc.*, the court reversed a summary judgment against a defendant that developed a competing program that copied the words and structure of Lotus's menu command hierarchy in its spreadsheet program (but not the underlying source code). 49 F.3d 807 (1st Cir. 1995), *aff'd by an evenly divided court*, 516 U.S. 233 (1996). The court found the menu command hierarchy to be functional and uncopyrightable. *Id.*

In so holding, the court concluded that "the Lotus menu command hierarchy is an uncopyrightable 'method of operation.'" *Id.* at 815. The court reasoned that it would be "absurd" to require every software provider to create different methods for performing each function in a program, which would require users to learn a multitude of different ways to, for example, print

files. *Id.* at 817-18. "The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable." *Id.* at 818; *see also* 17 U.S.C. § 102(b) (method of operation is unprotectable "regardless of the form in which it is described, explained, illustrated, or embodied").

The same reasoning applies to APIs for a programming language. *See* Astrachan Decl., Ex. 1 ¶¶ 91-98 (explaining why the Java language APIs are methods of operation). By providing Android API packages addressing many of the same common methods that are provided by Oracle's Java language API packages, Google has ensured that programmers using the Java language need not learn a new way to call these methods when writing programs for both the Java and the Android platforms. *See id.* ¶¶ 129-33. For example, to make use of the functionalities of the java.lang.Math class in the java.lang API package, a programmer would use the appropriate method name from the java.lang.Math class, such as "sqrt()," to calculate the square root of a number. *Id.* ¶ 25. When the program is run by the user, the underlying platform will then perform the "sqrt()" functionality and return the appropriate data. *Id.* This is no different from executing a menu command to perform a spreadsheet function. *See Lotus*, 49 F.3d at 816 ("If specific words are essential to operating something, then they are part of a 'method of operation' and, as such, are unprotectable."). Oracle has not even tried to allege that the underlying program logic in Android that *performs* this function was copied from the Asserted Works. *See supra*, Part II.D. It would be "absurd" to require that Android use, for example, different names than Oracle did for common mathematical methods, or for Android to group mathematical methods in different packages than Oracle did. "[F]orcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that 'methods of operation' are not copyrightable." *Lotus*, 49 F.3d at 818.

Similarly, in *Mitel, Inc. v. Iqtel, Inc.*, 896 F. Supp. 1050, 1055-56 (D. Colo. 1995), *aff'd*, 124 F.3d 1366 (10th Cir. 1997), the court held that command codes that served as a user interface by which technicians activated various functions of a call controller were unprotectable

methods of operation. "If, arguably, the command codes are considered part of the computer program in the call controller then their sole purpose is to provide access to the functions available in the call controller. Thus, they provide the means to access or operate the program contained in the software." *Id.* at 1055. Like the command codes in *Mitel*, the Java language APIs at issue constitute unprotectable methods of operation. At their most abstract level, their purpose is to provide access to common functions of language that are frequently used by programmers consistent with a published and familiar Java language API specification. *See* Astrachan Decl., Ex. 1 ¶¶ 129-33.

The purpose of the Copyright Act is "[t]o promote the Progress of Science and useful Arts . . . ." U.S. CONST. art. I, § 8, cl. 8. To accomplish this goal, others must be allowed "to build freely upon the ideas and information conveyed by a work." *Feist*, 499 U.S. at 350. As the *Lotus* court recognized, "[i]n the context of methods of operation, . . . 'building' requires the use of the precise method of operation already employed; otherwise, 'building' would require dismantling, too." 49 F.3d at 818; *see also Sega*, 977 F.2d at 1522 (citing § 102(b) for the proposition that "functional requirements for compatibility" are not protected by copyright).[16] It is undisputed that the *program logic* (other than elements of the 12 files addressed below in Part III.B) for the Accused Packages was not copied. Because the API *specifications* are methods of operation, they are not protected by copyright.

2. The API declarations are unprotectable *scenes a faire* or unprotectable under the merger doctrine.

The allegedly copied elements of the Java language API packages are also not protectable because they are required for interoperability; they allow a program written in the Java programming language to run on the Android platform. Expression dictated by external factors, such as industry practices and standard techniques used to perform particular functions in a specific computing environment, is unprotectable *scene a faire* or unprotectable under the merger doctrine, which precludes copyright protection for expression in which the underlying

[16] Notably, in other products, Sun and Oracle have implemented other companies' APIs in order to increase compatibility of the Sun and Oracle products with those of those other companies. *See* Astrachan Decl., Ex. 1 ¶¶ 62-90.

idea and the expression merge. *See Computer Assocs. Int'l, Inc. v. Altai*, 982 F.2d 693, 709-10 (2d Cir. 1992).

Because computer programs are essentially "utilitarian" in nature, *see Sega*, 977 F.2d at 1524, these external factors are prominent and will render unprotectable elements of a program dictated by compatibility[17] requirements, industry demands, or widely accepted programming practices. *Sega*, 977 F.2d at 1524. This is because "in many instances it is virtually impossible to write a program to perform particular functions in a specific computing environment without employing standard techniques." *Computer Assocs.*, 982 F.2d at 709 (quoting 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[F][3]); *see also id.* at 708 (in view of desire to create programs that meet users' needs efficiently, "the applicability of the merger doctrine to computer programs becomes compelling"); *Lexmark Int'l., Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 535-36 (6th Cir. 2004) (programming efficiency "figures prominently in the copyrightability of computer programs").

Courts have routinely emphasized that compatibility is a legitimate aim, and can override claims for infringement under either the *scenes a faire* or fair use doctrines. *E.g., Bateman*, 79 F.3d at 1547; *Baystate Techs. v. Bentley Sys.,*, 946 F. Supp. 1079, 1087-90 (D. Mass. 1996); *Mitel*, 896 F. Supp. at 1055-56. In *Baystate*, the plaintiff sued Bentley, a competing CAD program developer, contending that Bentley had infringed Baystate's rights in the CADKEY source code, the Part File Tool Kit source code and the Part File Tool Kit documentation. 946 F. Supp. at 1085-86. The Part File Tool Kit consisted of definitions or header files, a library of executable files, and documentation that described the organization of the file data structure and descriptions of the access functions included in the library of executable files, and allowed a user or third party developer to read and write the data files into a non-CADKEY program. *Id*. at 1083. Bentley's contractor admitted that it "referred" to the Tool Kit documentation in

[17] "[I]f manufacturer 'B' would like to design a computer that can use software designed for a computer manufactured by 'A,' *B's operating system interface must match that of manufacturer A.* Without that compatibility, the software designed for A's machine will not function on B's." *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1546 (11th Cir. 1996); *see also id.* at 1537 n.11 (explaining key terms and concepts involved in software copyright cases, including "operating systems," "application programs," and "interfaces").

developing a data translator that would permit the transfer of information between Baystate's CADKEY program and Bentley's CAD program. *Id*. at 1082-87. The court held that the names and organization of the data structures were not protectable and there was no infringement:

> The product being developed is a data translator that is designed to "read" the data files of CADKEY. The process of "reading" the CADKEY data files requires that the elements contained within the data structures of the Translator be organized in the same manner as the elements in the data structures of CADKEY. Without such compatibility, the Translator would not function because it would "misread" the CADKEY data files.

Id. at 1088.

Similarly, the court in *Mitel* found the command codes at issue to be unprotectable *scenes a faire*. The court found that the "proclivities of technicians largely dictate the need to conform the command codes in order to have market accessibility." *Mitel*, 896 F. Supp. at 1055-56. Because Iqtel had copied the command codes for efficiency reasons (as "[t]echnicians who installed and programmed call controllers were accustomed to Mitel's command codes and expressed a reluctance to learn new codes"), use of those elements was dictated by external factors. *Id*. at 1053.

The similarities in the unprotected elements present in the Oracle Java language API packages and the Android Java language API packages, *e.g.*, the names of packages and methods and definitions, are there for one purpose and one purpose only: enabling programs written in the Java programming language to run on the Android platform. *See* Astrachan Decl., Ex. 1 ¶¶ 129-33. If different names had been used for these interfaces in Android, existing code written in the Java programming language simply would not run properly and the programs would not function, and programmers writing new applications in the Java language for Android devices would have to learn new API names for methods they already knew. *See id.* ¶ 135. As in *Baystate*, where the names and organization of the data files used by the defendant had to be identical to those in plaintiff's program in order for defendant's translator to function correctly, the specifications for the APIs must be identical in order for programs written in the Java programming language to operate on Android devices. *See also* Paul GOLDSTEIN, GOLDSTEIN ON COPYRIGHT § 2.15.3, at 2:208 (3d ed. 2011) (identifying machine to machine, program to

program, and program to machine interface specifications as interfaces "in which design consumes relatively few creative resources and no room for variation exists if one element is to be able to interoperate with the other").

The APIs at issue were implemented in order to provide experienced programmers with an efficient, familiar method of performing the same basic operations. *See* Astrachan Decl., Ex. 1 ¶¶ 134-38 (discussing industry demand for compatibility). For example, as has already been noted, for compatibility and standardization reasons, the "abs" function discussed earlier has the following identical method declaration not only in Java and Android, but also in Harmony and GNU Classpath. *Id.* ¶ 61. Indeed, both Harmony and GNU Classpath have implemented the API packages that are the subject of Oracle's infringement claim against Google. In each case, the names of the packages and methods are the same, *id.*, so that programmers need not resort to the "absurd" task of learning to perform the same operation in different ways. While the program logic that performs the actions required by the APIs may be (and in fact is) different, the names (or abbreviations), specifications for the APIs must be identical to allow for interoperability. *Id.* ¶ 53.

3. The Java language API package and method names are unprotectable as a matter of law.

Oracle has alleged that Google "copied" the names of the Java language APIs at issue. But the names of the Java language API files, packages, classes, and methods are not protectable as a matter of law. *See* 37 C.F.R. § 202.1(a) ("Words and short phrases such as names, titles and slogans" are not subject to copyright). Moreover, the names merely describe the functionality of the packages and methods, and/or otherwise are the result of customary programming practices. Astrachan Decl. ¶ 7(b)-(f) & Ex. 1 ¶¶ 99-117, 120-22; *see also Merchant Transaction Sys., Inc. v. Nelcela, Inc.*, No. CV 02-1954-PHX-MHM, 2009 WL 723001, at *12 (D. Ariz. March 18, 2009) ("Clearly, isolated field names 'are not original, are within the public domain, and are not entitled to individual protection.'"); *CMM Cable Rep, Inc. v. Ocean Coast Props.*, 97 F.3d 1504, 1519 (1st Cir. 1996) ("It is axiomatic that copyright law denies protection to 'fragmentary words and phrases' and to 'forms of expression dictated solely at functional considerations' on the

grounds that these materials do not exhibit the minimal level of creativity necessary to warrant copyright protection."); *Baystate*, 946 F. Supp. at 1088 (finding that "data structure names or, more specifically, the words and abbreviations used to describe the files contained within the data structures and the data structures themselves" were not protected by copyright because "the name of a file is typically related to its function").

The names, methods, and contents of the Java language API packages reflect customary naming practices in the computer industry. For example, package names such as "java.io," "java.util," and "java.net" reflect industry shorthand for common functionality, namely input/output, utilities, and networking, respectively. Astrachan Decl., Ex. 1 ¶ 124. These packages with common names then, in turn, contain methods whose naming reflects industry custom and representation of the underlying functionality. Astrachan Decl. ¶ 7 (d)-(f) & Ex. 1 ¶¶ 103-17. Some examples of function names that are very similar in Java and the pre-existing C and C++ languages as a result of their functionality and industry custom include: "char" to identify a data type holding a character; "int abs (int i)" to identify a function that returns the absolute value of a number; and "printf()" for a function that prints a string to an output device. Astrachan Decl., Ex. 1 ¶ 124. All of these names were used in at least the C language before they were used in Java. *Id.*; *see also* Astrachan Decl., Ex. 2 at 53 ("Java feels very familiar to many different programmers because Sun had a very strong tendency to prefer things that had been used a lot over things that just sounded like a good idea.").

These are not the only examples; many of the "names" used in the Java language have been used by the industry for decades. For example, the C Reference Manual, written by Brian Kernighan, the inventor of the C programming language, and published in 1975, references "int," "double" and "char"—all of which are used in Java. Astrachan Decl., Ex. 1 ¶ 125. "Bool" (which became "Boolean" in Java) dates back further, to at least 1968. *Id.; see also id.* ¶¶ 126-28; *Computer Assocs.*, 982 F.2d at 710 ("Closely related to the non-protectability of *scenes a faire*, is material found in the public domain. Such material is free for the taking and cannot be appropriated by a single author even though it is included in a copyrighted work," citing *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1473 (9th Cir. 1992), in affirming the district

court's finding that "'[p]laintiffs may not claim copyright protection of an . . . expression that is, if not standard, then commonplace in the computer software industry'").

4. Alternatively, any similarity between the works is a fair use.

For much the same reasons that the elements of the Java language APIs at issue are unprotectable, the inclusion of those elements in Android for the purpose of allowing third party Java programmers to write compatible applications constitutes a fair use as a matter of law. *See Bateman*, 79 F.3d at 1547 (relying on fair use as an alternative basis for non-infringement); *Mitel*, 896 F. Supp. at 1056 ("Even assuming the command codes are copyrightable, Iqtel's use of these same command codes constitutes a 'fair use' under the Act.") A finding of fair use can be made on summary judgment and is especially appropriate where, as here, any alleged copying serves a transformative purpose. *E.g.*, *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 817-22 (9th Cir. 2003) (affirming summary judgment finding that defendant's use of thumbnails that copied plaintiff's images and linked to the website where the image was stored was fair use).

The use of the same names and functional method declarations for the thirty-seven Java language APIs serves the transformative purpose of allowing applications written in the Java language to run on the Android platform. Section 107 of the Copyright Act lists several non-exhaustive factors to be considered to determine fair use: (1) the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect of the use upon the potential market for or value of the copyrighted work. *Sega*, 977 F.2d at 1521-22.

The character of the use: Where, as here, a use serves a transformative purpose, summary judgment is appropriate. *See, e.g.*, *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146, 1164 (9th Cir. 2007) ("A work is 'transformative' when the new work does not 'merely supersede the objects of the original creation' but rather 'adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message,'" citing *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 579 (1994)).

As discussed above, the names and method declarations of the Java language API

packages are used in Android for compatibility and standardization reasons, as part of the open source Android platform. Open source licensing has generally transformed what have historically been thought of as commercial uses, creating a new model of technology development that leverages public goods and participation. *See Sega*, 977 F.2d at 1523 (courts "are free to consider the public benefit resulting from a particular use"; "[p]ublic benefit need not be direct or tangible, but may arise because the challenged use serves a public interest"; finding fair use and concluding that "Accolade's identification of the functional requirements for Genesis compatibility has led to an increase in the number of independently designed video game programs offered for use with the Genesis console").

Moreover, the Ninth Circuit has specifically held that a computer program is transformative if it is developed for a new platform, even one that competes with that of the copyright owner. *See, e.g., Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 606-07 (9th Cir. 2000). The court in *Sony* held it was a fair use for Connectix to reverse engineer the BIOS software that operates the Sony PlayStation console in order to create a competing program that emulated the functioning of the PlayStation and allowed PlayStation games to be played on a computer. *Id.* at 599, 608. In finding fair use, the *Sony* court concluded that the defendant's product was transformative because it created a new product that was compatible with PlayStation games, "notwithstanding the similarity of uses and functions between the Sony PlayStation" and Connectix's product. *Id.* at 606-07. Similarly, Google's implementation of the Java language APIs at issue was transformative because it enabled programmers to develop applications written in the Java programming language to run on Android, a wholly new "smart phone" platform.

The nature of the copyrighted work: "To the extent that a work is functional or factual, it may be copied." *Sega*, 977 F.2d at 1524, 1527 (concluding that computer programs are "essentially utilitarian" in nature and that under the Copyright Act, "if a work is largely functional, it receives only weak protection"). As discussed above, the APIs at issue serve an entirely functional purpose and were included in Android to enable interoperability.

The portion of the work used: The thirty-seven API packages identified by Oracle are

approximately one-quarter of the API packages in the Asserted Works. Within these packages and their implementations, only the names of the packages and methods were used by Google for compatibility reasons. It is undisputed (and Oracle has never alleged to the contrary) that the underlying source code *implementing* those packages and methods, which constitute the vast majority of that code—hundreds of thousands of lines, *see* Astrachan Decl. ¶ 7(h) & Ex. 1 ¶ 143—was not used by Google. Instead, Google independently developed or obtained from a third party open source project the code that implemented these APIs. Because Google used only the portions of the API packages necessary for compatibility, this factor weighs in its favor. *See Kelly*, 336 F.3d at 820-21 ("If the secondary user only copies as much as is necessary for his or her intended use, then this factor will not weigh against him or her.").

The effect upon the market for the copyrighted work. If anything, Android has contributed positively to the market for the copyrighted works by increasing the number of Java language developers. As Sun's CEO Jonathan Schwartz said when Google released the Android SDK in 2007, "needless to say, Google and the Open Handset Alliance just strapped another set of rockets to the [Java] community's momentum - and to the vision defining opportunity across our (and other) planets." Kwun Decl., Ex. L. Rich Green, then Sun's executive vice president of software, also said that Sun was "thrilled to have Google amplify the global momentum behind Java technology . . . ." *Id.*, Ex. M.

Moreover, even if Android's use of these API elements was considered to be competitive with Oracle's, the outcome is the same. *E.g.*, *Sega*, 977 F.2d at 1523 (Accolade "sought only to become a legitimate competitor in the field of Genesis-compatible video games. Within that market, it is the characteristics of the game program as experienced by the user that determine the program's commercial success. As we have noted, there is nothing in the record that suggests that Accolade copied any of those elements."); *Kelly*, 336 F.3d at 821 ("A transformative work is less likely to have an adverse impact on the market of the original than a work that merely supersedes the copyrighted work"); *Sony*, 203 F.3d at 607 (recognizing that some economic loss by Sony as a result of the defendant's competition does not compel a finding of no fair use; "Sony understandably seeks control over the market for devices that play games

Sony produces or licenses. The copyright law, however, does not confer such a monopoly.").

Finally, like the uses of these same common elements by others such as Apache Harmony and GNU Classpath, Android's use of the Java language API names and method descriptions are essential for compatibility. Their inclusion in Android could not have had any greater impact than the existing use of these same functional elements by others already in the "market."

Google's use is a fair use. The limited and transformative nature of Google's use of the functional names and method descriptions of the APIs to make Android's mobile platform interoperable and compatible with Java applications and to further the efforts of third party developers weighs heavily in favor of finding fair use. "In determining whether a challenged use of copyrighted material is fair, a court must keep in mind the public policy underlying the Copyright Act. . . . [T]he fundamental purpose of the Copyright Act [is] to encourage the production of original works by protecting the expressive elements of those works while leaving the ideas, facts, and functional concepts in the public domain for others to build on." *Sega,* 982 F.2d at 1527; *see also Sony*, 203 F.3d at 608 ("The four statutory fair use factors must be 'weighed together, in light of the purposes of copyright,'" citing *Campbell*, 510 U.S. at 578); *Perfect 10*, 508 F.3d at 1166, 1168 ("'the more transformative the new work, the less will be the significance of other factors, like commercialism, that may weigh against a finding of fair use,'" quoting *Campbell*, 510 U.S. at 579; finding fair use where Google's thumbnails were transformative and there was no proof of harm to Perfect 10's market); *Kelly*, 336 F.3d at 818 (applying same principle).

B. The alleged similarities in the remaining 12 files are *de minimis* in the context of the over *9,500* files in the Asserted Works, and the over 50 *thousand* files in Android.

De minimis acts of copying are not actionable. *Newton v. Diamond*, 388 F.3d 1189, 1192-93 (9th Cir. 2004). Where the only similarity is as to "nonessential matters," the copying is *de minimis*. *See id.* at 1195 (quoting 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[A][2], at 13-48). Where a defendant copies only "a portion of the plaintiff's work exactly or nearly exactly . . . the dispositive question is whether the copying goes to trivial or substantial elements." *Id.* That substantiality is judged by "considering the qualitative and

quantitative significance of the copied portion in relation to the *plaintiff's work as a whole*." *Id.* (emphasis added).

The twelve Accused Files that Oracle contends contain Oracle source code or other material allegedly copied from the Asserted Works cannot support a finding of infringement. First, even assuming for purposes of this motion that the Accused Files contain copied Oracle material, that copying is quantitatively miniscule, comprising less than 750 lines of comments and code out of 2.8 *million* lines of code in the Asserted Works as a whole, and 12 files out of more than *9,500* files in the Asserted Works as a whole. Astrachan Decl., Ex. 1 ¶ 150.

Second, the allegedly copied lines are qualitatively insignificant. Eight of the files (files provided by an independent contractor to Google, which have since been removed from the Android platform) are simple test files that do not have any effect on the code that ships on Android devices, and are never present on Android devices. Bornstein Decl. ¶ 8; Astrachan Decl., Ex. 1 ¶¶ 160-68. In two other files, also provided by the independent contractor, the allegedly copied lines are only in the *comments* (which also have since been removed), and those comments are in any event are largely descriptive and functional. Bornstein Decl. ¶ 9; Astrachan Decl., Ex. 1 ¶¶ 169-77. As to these two files, any alleged copying has no effect on the code in shipped devices because comments do not change the code that results from a programming file. Astrachan Decl., Ex. 1 ¶ 171. The final two files both contain, as part of hundreds of lines of code, the same nine lines similar to code in the Asserted Works. *Id.* ¶ 152. Those nine lines[18] implement a mundane task that, far from being the heart of the Asserted Works,[19] represent a utilitarian function that performs a simple "sanity check" to make sure that certain variables are "in bounds." *Id.* ¶¶ 153-66; *see Newton*, 388 F.3d at 1196 (expert testimony that the copied elements were "common, trite, and generic" sufficed to demonstrate that the elements were qualitatively insignificant).

Because the alleged similarities in the twelve files are both quantitatively and

---

[18] *See* Astrachan Decl., Ex. 1 ¶ 152 (nine lines of code at issue).

[19] Where the defendant copies the "heart" of the work, a relatively smaller taking, quantitatively speaking, may exceed *de minimis* use. *Fisher v. Dees*, 794 F.2d 432, 434 n.2 (9th Cir. 1986).

qualitatively insignificant, any alleged copying is *de minimis*, and thus not actionable. *MiTek Holdings, Inc. v. Arce Eng'g Co., Inc.*, 864 F. Supp. 1568, 1585 (S.D. Fla. 1994) ("Of the protectable elements that are substantially similar, the Court has found that their lack of importance in the context of the programs as a whole renders any copying by Arce to be *de minimis*."), *aff'd*, 89 F.3d 1548, 1560 (11th Cir. 1996) ("We agree with the district court that the elements that were considered original and appropriated were not of such significance to the overall program to warrant an ultimate finding of substantial similarity and hence infringement.")

C. The documentation for the Android APIs is not substantially similar or virtually identical to the documentation for the Java language APIs.

Any similarities between the machine-generated documentation for the Android APIs and that of the Java language APIs are due to either the similarities in the APIs themselves, or industry practices regarding how documentation is written. *See* Astrachan Decl., Ex. 1 ¶¶ 145-49. Just as two dictionaries will often have similar sounding definitions both because they are defining the same thing and due to external constraints (such as brevity and clarity) on how dictionary definitions are written, skilled technical writers writing about the same API are likely to come up with descriptions that appear very similar and contain very similar descriptions of the critical features. *Id.* ¶¶ 146-48. Given the substantial unprotected elements in the documentation (such as the API method declarations), the "virtual identity" standard applies here. *Apple*, 35 F.3d at 1439. No reasonable jury could find infringement based on the Android API documentation.

D. Any claim based on Oracle's selection and arrangement of allegedly copied elements must be evaluated under the "virtual identity" standard, and Oracle cannot establish infringement under that standard.

Finally, any residual claim of infringement based on the selection or arrangement of allegedly copied elements must also fail. Here, the method declarations, package names, and the rest of the API specifications are unprotectable, as discussed above. *See also* Astrachan Decl., Ex. 1 ¶¶ 48-51, 118-19. Any remaining allegedly copied material (i.e. the portions of the twelve Accused Files) is, as explained above, *de minimis*. On these facts, "to the extent there is creative

expression left in how the works are put together, as a whole they can receive only limited protection." *Apple*, 35 F.3d at 1439. This means that "illicit copying could occur only if the *works as a whole* are virtually identical." *Id.* at 1447 (emphasis added).

Far from being virtually identical to the Asserted Works as a whole, the vast majority of the Android platform is *different* from the Java platform. Oracle has not even tried to claim that the Android APIs derive from more than about one-quarter of the API packages in the Asserted Works. Moreover, Android includes a host of elements, such as the Dalvik VM and a Linux kernel, that are not part of the Java platform. Astrachan Decl., Ex. 1 ¶¶ 140-42; *see also* Astrachan Decl. ¶ 7(h)-(i) & Ex. 1 ¶¶ 143-44. Because the "works as a whole" are not virtually identical, there is no copyright infringement.

E. Oracle's secondary infringement claims fail for lack of proof of direct infringement.

Oracle cannot establish direct infringement for the reasons detailed above. Its inducement and contributory infringement claims must therefore also fail. *Subafilms, Ltd. v. MGM-Pathe Communications Co.*, 24 F.3d 1088, 1092 (9th Cir. 1994); *Matthew Bender & Co. v. West Publ'g Co.*, 158 F.3d 693, 706 (2d Cir. 1998).

IV. CONCLUSION

There are no genuine disputes of material fact relating to the controlling legal issues; Google is entitled to judgment as a matter of law. As Sun's CTO testified to Congress in 1994, the lack of copyright protection for interface specifications drives "innovation, competition, and economic investment" and "[a]rguments to the contrary mix up the distinction between interface specifications and product implementations, in an attempt to retain or regain monopoly control to limit competition." *See* Kwun Decl., Ex. G at 2. Google respectfully requests that the Court grant it summary judgment on Oracle's Count VIII, for copyright infringement.

Dated: August 1, 2011

KEKER & VAN NEST LLP

By: s/ Michael S. Kwun
MICHAEL S. KWUN
Attorneys for Defendant
GOOGLE INC.